An integrated approach to concept recognition in biomedical text

William A. Baumgartner, Jr.[1] **Zhiyong Lu**[1] **Helen L. Johnson**[1]
J. Gregory Caporaso[1] **Jesse Paquette**[1] **Anna Lindemann**
Elizabeth K. White **Olga Medvedeva** **K. Bretonnel Cohen**[1]
Lawrence Hunter[1]

[1] Center for Computational Pharmacology, University of Colorado School of Medicine

Abstract

Our approach to the three BioCreative 2006 tasks had three main characteristics: (1) Extensive use of UIMA (Unstructured Information Management Architecture), a framework that facilitates integration and evaluation of system components, as well as incorporation of third-party tools. (2) Extensive use of a semantic parser, OpenDMAP (Open Source Direct Memory Access Parser). (3) Use of domain-specific rule-based approaches for handling coordination of protein names. We noted large differences between our performance on the training data and our performance on the test data in the IAS and IPS sub-tasks.

Keywords: semantic parsing, conceptual language processing, knowledge-based language processing, direct memory access parsing (DMAP)

1 Introduction

The approach of the Center for Computational Pharmacology to the BioCreative 2006 tasks had three basic characteristics: (1) use of an architecture that allowed us to apply a single, integrated framework to all three tasks; (2) extensive use of a semantic parser; and (3) use of rule-based approaches to handling coordination of protein names.

We made extensive use of the UIMA (Unstructured Information Management Architecture) [11, 20] framework for integrating almost every component that we used in any BioCreative 2007 task. Three benefits accrued from this strategy: (a) The complete integration of all processing steps allowed us to quickly and easily experiment with different approaches to the many subtasks involved. (b) It made it easy for us to quickly evaluate the results of these experiments against the official data sets. (c) It provided us with a clean interface for incorporating tools from other groups, including LingPipe [4], ABNER [28], and Schwartz and Hearst's [27] abbreviation detection algorithm.

We also made extensive use of a semantic parser being developed by our group. Called OpenDMAP (Open source Direct Memory Access Parser), it is a modern implementation of the DMAP paradigm first developed by Riesbeck [26], Martin [21], and Fitzgerald [12]. The earliest descriptions of the paradigm assumed that a DMAP system would approach all levels of linguistic analysis, from part-of-speech choice through word sense discrimination to extraction of propositional content, through a single optimization procedure. In this work, we show that analysis can be modularized, and even externalized, without losing the essential semantic flavor of the DMAP paradigm.

Finally, we developed a number of rules for handling the domain-specific conjunction strategies of biomedical text, such as using *BMP1/2* to mean "BMP 1 and BMP 2."

2 Gene Mention Task

Our system for the 2006 Gene Mention (GM) task focuses on simple consensus approaches for combining the output of multiple gene taggers. We used three taggers: an in-house tagger developed for the BioCreative 2004 gene mention task (Task 1A) [16] and two publicly available taggers, ABNER

[28] and LingPipe [4]. Integration of each tagger into the system was accomplished using the UIMA [11, 20]. Our overall consensus approach can be divided into two general strategies which test two distinct hypotheses. *Hypothesis #1* poses that filtering the output of multiple gene/protein mention identification systems by requiring agreement by two or more of the individual systems will result in a precision measure greater than or equal to the highest precision measure of the individual components. *Hypothesis #2* states that combining the output of multiple gene/protein mention identification systems will result in a recall measure greater than or equal to the highest individual recall measure of the individual components.

To test these hypotheses, we implemented two filter varieties which combine the output of gene taggers in different ways. It should be noted that the taggers used for the GM task were used "out-of-the-box," that is, they were not trained on the BioCreative 2006 data. The models used for each tagger were trained on data from the inaugural BioCreative gene mention task, and judging from the results, each tagger was trained on different parts of the original data (Table 1).

Consensus Filter: To test Hypothesis #1, we developed a consensus filter, analogous to a voting scheme. Each tagger votes, and a gene mention is kept if it accumulates a certain threshold of votes. If the threshold is not met, the gene mention is removed from the analysis. We used two consensus approaches, one which required two of the three taggers to agree, and the other which required unanimous agreement in order for a gene mention to be kept. By combining the output of three taggers which are known to have decent performance on their own, we expected that the consensus approach would result in an elevation in overall precision, without dramatically decreasing recall. Although it might seem intuitively sensible to weight the vote of each tagger, perhaps by the performance data reported in Table 1, that data is not clearly directly comparable, since each tagger was trained on different subsets of the 2004 data. Therefore, we weighted each tagger's vote equally.

Overlapping Filter: To maximize recall (and test Hypothesis #2), we implemented a simple filter which keeps all gene mentions by resolving overlapping mentions among the taggers. When an overlap between two gene mentions is detected, the filter compares their respective span lengths, and keeps the gene mention with the greater span[1]. By keeping all gene mentions, we expected to increase the recall of the system; however, we also expected the precision of the system to suffer, since more false positives will be returned.

Table 1: Performance of the individual gene taggers on the 2006 training data broken down according to the 2004 BioCreative data sets.

	2004 Test			2004 Train			2004 Dev		
Tagger	P	R	F	P	R	F	P	R	F
CCP	77.5	77.9	77.7	88.5	88.6	88.6	81.2	79.3	80.2
ABNER	78.0	73.7	75.8	89.2	89.0	89.1	78.0	70.7	74.2
LingPipe	88.1	92.6	90.3	88.6	92.9	90.7	88.5	92.5	90.5

Results: We conducted a simple experiment to gauge the differences in training data used for each of the three taggers. Table 1 shows the performance of each tagger on the 2006 data. The data has been divided according the the three different data sets provided in the inaugural gene mention task, *test*, *train* and *dev*. Having constructed the CCP tagger in-house, we know that it was trained on the *train* and *dev* portions of the data which is reflected in the performances depicted in the table. The results of this experiment suggest that our implementation of ABNER was trained on only the *train* data, while the LingPipe model used was generated using all three subsets of the data.

As expected the consensus approaches increased precision over the individual tagger performances for the training data (See Table 2). For the overlapping filter, we actually note a worse recall than for

[1] An alternative would be to return the shortest overlapping span; having noted that BioCreative 2004 Task1A systems that took steps to extend multi-word name boundaries rightward and leftward benefitted from doing so, we chose the longer span.

Table 2: Performances of systems and individual components on the 2006 test and training data. Median score, as supplied by organizers. Quartiles for our runs are shown in parentheses.

	Test Data			Training Data		
Tagger	Precision	Recall	F-measure	Precision	Recall	F-measure
CCP	77.30	77.74	77.52	83.68	83.48	83.58
ABNER	80.38	73.26	76.65	83.85	80.86	82.33
LingPipe	72.53	80.00	76.09	88.47	92.77	90.57
2/3 Majority	85.54 (2)	76.83 (3)	80.95 (3)	91.15	86.33	88.68
Unanimous	92.78 (1)	49.12 (4)	64.24 (4)	94.56	61.41	74.46
Overlap	66.22 (4)	83.72 (2)	73.94 (4)	79.27	91.17	84.80
Median	85.08	79.05	81.32			

LingPipe individually, however, since LingPipe appears to have been trained on the entire data set, it is reasonable to expect the overlapping filter to have a lower recall in this case.

The test data provides a more accurate testing ground for our hypotheses. As with the training data, the consensus approaches are observed to elevate precision over any of the individual components. The overlapping filter also behaved as expected, by increasing the system's overall recall measure, however the dramatic decrease in precision is an unfortunate side effect.

3 Gene Normalization Task

Most aspects of our approach to the GN task are fairly conventional: we identify gene mentions; normalize typographic and morphological variants; look for a unique Entrez Gene entry to map to; if multiple entries are found, disambiguate between them. The primary novelty of our approach lies in the steps that we take to deal with conjunction.

Gene mention localization: In the gene mention localization step, we focussed on maximizing recall, while taking very basic steps to avoid false positives. To maximize recall, we applied six separate GM systems([14],[29],[16],[4], and [28] with the BioCreative 2004 model and the NLPBA model) and the *overlapping filter* described in Section 2 above. After manually examining false positives output by this system when run on the training data, we developed a set of 9 heuristics (listed in Table 3) to filter out obvious false positives and implemented them using regular expressions. Application of all 9 heuristics resulted in removal of 1086 putative gene mentions, and an increase of precision from 0.770 to 0.829 and of recall from 0.673 to 0.725 on the GN task.

Table 3: Effects of distractor string removal rules on GN scores. Step 0 means no preprocessing steps applied. At each step, the preprocessing rules that precede it are also applied. *Removed* refers to the cumulative number of gene mentions removed.

	Removal of ...	Example	P	R	F	Removed
0			0.770	0.673	0.718	0
1	gene chromosome location	*3p11-3p12.1*	0.772	0.673	0.719	34
2	single, short lowercase word	*heme*	0.778	0.672	0.721	112
3	strings of only numbers &/or punct	*9+/-76*	0.779	0.672	0.722	206
4	extra preceding words	*protein SNF* to *SNF*	0.790	0.681	0.731	225
5	extra trailing words	*SNF protein* to *SNF*	0.812	0.723	0.765	419
6	amino acids	*Ser-119*	0.815	0.723	0.766	460
7	protein families	*Bcl-2 family proteins*	0.816	0.722	0.766	701
8	protein domains, motifs, fusion	*SNH domain*	0.828	0.722	0.771	883
9	non-human proteins	*rat IFN gamma*	0.829	0.725	0.774	1086

Conjunction resolution: We noted that approximately 8% (52/640) of gene names in the development data set contained conjunctions—either General English ones, e.g. *HMG1 and 2,* or domain-specific ones, e.g. *IL3/5.* We developed a procedure for extracting individual gene names from conjunctions of the following types:

1. Gene names in regular coordinated structures (e.g. *IL3/IL5* refers to *IL3* and *IL5*).
2. Individual gene names in a series omitted (e.g. *freac1-freac7* refers to *freac1*, *freac2*, *freac3*, *freac4*, *freac5*, *freac6* and *freac7*).
3. Gene subtypes separated after the main gene name (e.g. *IL3/5* refers to *IL3* and *IL5*).
4. Gene subtypes separated before the main gene name (e.g. *M and B creatine kinase* is transformed to *M creatine kinase* and *B creatine kinase*).

The algorithm first looks for two typical conjunction-indicating words: *and* and *to*—and two atypical, domain-specific conjunction-indicating forms: *forward slash (/)*, and *hyphen (-).* Then the algorithm builds the individual gene names from the conjoined structure. (See [19] for further details.)

Table 4 shows the overall improvement in performance on the training data yielded by the conjunction resolution step. It is slight—F-measure increases only from 0.763 to 0.777, even though as we pointed out above, about 8% of gene tokens in the data appear in structures requiring some processing. One reason for this is that some of the conjoined genes are also mentioned individually, allowing for their normalization without having to handle the conjunction. Another reason is that some conjunctions were beyond the scope of our algorithm, e.g. *granulocyte (G-) and granulocyte-macrophage (GM-) colony-stimuating factor (CSF)).*

Table 4: GN results on the training data with and without conjunction resolution.

Steps	Precision	Recall	F-measure
without conjunction resolution	0.836	0.691	0.757
with conjunction resolution	0.827	0.727	0.774

Regularization of typographic and morphological variants: We built a dictionary of gene names and symbols, and then used a set of heuristics to regularize all gene mentions in the dictionary and in the output of the GM step.

Dictionary construction: We extracted the gene symbol, synonyms, and full name from Entrez Gene. In addition to Entrez Gene [2], we also investigated other databases such as UniProt [3] and a combination of the two databases. We found the Entrez Gene database to be the best resource for gene dictionary construction for the current task (See Table 5). This result is consistent with the conclusions reported in [7].

Examination of the dictionary entries showed that some entries could be removed without adversely affecting system performance because they are of no use for gene normalization tasks. Four pruning rules were implemented to facilitate their removal. Gene entries that begin with "LOC" or were preceded by "similar to" are temporary and often become discontinued in Entrez Gene, and are therefore unlikely to appear in text. Genes that were classified as either "hypothetical" or as a "pseudogene" were also excluded. These four classes of gene names were removed from the dictionary as it has been shown that smaller gene dictionaries have advantages over larger dictionaries [34]. However, it should be noted that removal of these four gene name classes had no impact on system performance. The dictionary used for the GN task contained 21,206 gene entries; see Table 6 for details.

Gene mention regularization: We then used a set of heuristics to regularize all gene names and symbols in the dictionary and in the output of the GM step. These heuristics are based on our

[2] Homo_sapiens.ags.gz file available at `ftp://ftp.ncbi.nih.gov/gene/`

[3] `http://www.uniprot.org`

Table 5: GN results on the development set using different online resources for lexicon construction.

Resources	Genes Entries	Precision	Recall	F-measure
Entrez Gene	21,206	0.827	0.727	0.774
UniProt	18,580	0.834	0.591	0.692
EG + UniProt	24,182	0.827	0.708	0.762

Table 6: The number of Entrez Gene entries removed in the *Homo_sapiens.ags.gz* file downloaded on 16 August 2006 according to the filtering rules. Rules are applied in order.

	Matching Term	Removed Gene Entries	Remaining Gene Entries
1	*LOC\d+*	14,831	24,273
2	*similar to*	86	24,187
3	*hypothetical*	264	23,923
4	*pseudogene*	2,717	21,206

own early work and on previous dictionary-based systems [9, 7, 10]. Table 7 shows the effects of the individual rules on performance. In particular, transformation rules for case normalization and space removal played roles in improving recall; the last rule for removing very short strings enhanced precision by quite a large margin. Use of all seven rules, in sequential order, resulted in an increase of F-measure from 0.586 to 0.774.

Mapping mentions to Entrez Gene IDs: After the extracted gene mentions have been regularized, and conjuctions have been addressed, the processed mentions are compared to all entries in the dictionary using exact string matching. If there are multiple matches, then all of the matched entries are taken into the disambiguation step discussed below. After the extracted gene mentions have been normalized, and conjuctions have been addressed, the processed mentions are compared to all entries in the dictionary using exact string matching. Three outcomes are possible:

1. If there is no match, then nothing is returned.
2. If there is a single match, then it is returned.
3. If there are multiple matches, then all of the matched entries are taken into the disambiguation step discussed below.

In addition to exact string matching, we also investigated some approximate string matching techniques. Like [10], we found that approximate matching markedly increased search time but did not markedly improve performance.

Gene Name Disambiguation: For a given species, a gene name is *ambiguous* when it refers to more than one standard database identifier. For example, *CHED* is used as a synonym for two separate Entrez Gene entries: *CHED1* (GeneID: 8197) and *CDC2L5* (GeneID: 8621). It has been estimated that > 5% of terms for a single organism are ambiguous [32, 5] and that approximately 85% of terms are ambiguous across species. For the (single-species) GN task, we implemented two approaches to gene name disambiguation. The first method attempts to identify "definitions" of gene symbols, using the Schwartz and Hearst algorithm [27]. Our second approach is similar to that of ([17]), except that it uses the five tokens that appear before and after the ambiguous gene, rather than the entire sentence. In both cases, we calculate the Dice coefficient between the extracted text (abbreviation definitions or flanking tokens) and the full name of each gene candidate as given in Entrez Gene. (Our implementation of the Dice coefficient calculation uses a stop word list and stems each token [24]. The gene with the highest non-zero Dice coefficient is returned. If the Dice coefficients are all zero, we return nothing.

Our results indicate that finding unabbreviated gene names or flanking words plays an important role in resolving ambiguous terms (see Table 8). Moreover, this gene name disambiguation procedure can provide evidence for a term being a false gene mention. For example, *STS* (PMID: 7624774) is

Table 7: Heuristics used to normalize gene names in both lexicon construction and during processing of the gene tagger output, and the results after each step was performed. Step 0 means no string transformation was applied. At each rule, the processing rules that precede it are also applied.

	Rule	Example	P	R	F
0			0.783	0.469	0.586
1	Substitution: Roman letters >arabic numerals	*carbonic andydrase XI* to *carbonic andydrase 11*	0.778	0.492	0.603
2	Substitution: Greek letters >single letters	*AP-2alpha* to *AP-2a*	0.779	0.497	0.607
3	Normalization of case	*CAMK2A* to *camk2a*	0.787	0.619	0.693
4	Removal: parenthesized materials	*sialyltransferase 1 (beta-galactoside alpha-2,6-sialytransferase)* to *sialyltransferase 1*	0.782	0.623	0.694
5	Removal: punctuation	*VLA-2* to *VLA2*	0.768	0.667	0.714
6	Removal: spaces	*calcineurin B* to *calcineurinB*	0.784	0.742	0.762
7	Removal: strings <2 characters	*P*	0.827	0.727	0.774

Table 8: Results of gene normalization with and without disambiguation.

Steps	Precision	Recall	F-measure
without disambiguation	0.848	0.689	0.760
use abbreviations only	0.825	0.722	0.770
use abbreviations and flanking regions	0.827	0.727	0.774

recognized as a gene mention, but its surrounding words, *content mapping and RH analysis*, indicate it is an experimental method. We assembled a list of words suggesting non-protein terms such as *sequence* or *analysis*. When they were matched to a gene's unabbreviated name or its flanking words, the gene is considered as a false mention.

Even with the improvement yielded by disambiguation, ambiguity remains a contributor to system error: on the development data, our precision for mentions that only matched a single Entrez entry was 0.85, while for ambiguous entries, it was only 0.63. (Recall is difficult to differentiate for the two cases, since we do not know how many mentions in the gold standard are ambiguous.)

Other techniques applied: To further enhance system performance, especially in regard to false positive identification, we assembled a stop word list consisting of common English words, protein family terms, non-protein molecules, and experimental words, all of which are common distractor strings. The common English word stop list included 5,000 words derived by word frequency in the Brown corpus [13]. The protein family terms were derived from an in-house manual annotation project which annotated protein families. A list of small molecules, e.g. Ca2+, was also added.

Table 9: Results of gene normalization with different stop word lists.

Steps	Precision	Recall	F-measure
do not use stop list	0.764	0.739	0.752
use common English words stop list	0.776	0.738	0.757
use non-protein stop list	0.768	0.736	0.752
use custom stop list	0.811	0.730	0.769
use all three stop lists	0.827	0.727	0.774

Results on the test data: We submitted three separate runs for the GN task. Run 1 favored precision: it used all four stop lists and removed from the dictionary any terms that could be mapped to two or more identifiers. Run 2 aimed to optimize F-measure: it did not use the "protein family stop list," and removed terms associated with three or more database identifiers. Run 3 aimed to

Table 10: Evaluation results by our system on the development set are shown in the first three rows. Row 4 shows the estimated recall ceiling for lexical matching reported by [22] for the same data set.

Run	True Positives	False Positives	False Negatives	Precision	Recall	F-measure
1	458	94	182	0.830	0.716	0.769
2	465	97	175	0.827	0.727	0.774
3	467	103	173	0.819	0.730	0.772
4	530	7941	110	0.063	0.828	0.117

Table 11: Results on the GN test data.

Run	True Positives	False Positives	False Negatives	Precision	Recall	F-measure	Quartile
1	576	109	209	0.841	0.734	0.784	1
2	583	120	202	0.829	0.743	0.784	1
3	587	129	198	0.820	0.748	0.782	1

optimize recall: it used fewer stop lists, and removed terms from the dictionary only when they could be mapped to five or more database identifiers. Table 10 shows that the results do not vary widely from the development set. We were able to improve on the estimated recall ceiling for simple matching to a lexicon as reported in [22].

Table 11 shows results on the test data. F-measure for all three runs is in the top quartile and is comparable to the highest F-measure (0.79) for the GN task in mouse (the most comparable of the three species in BioCreative 2004).

We believe the most innovative components of our system are (1) the approach for handling complex coordination properly, and (2) the rules for disambiguating among multiple gene matches for a particular string. This system is also unusual for a rule-based dictionary method in that it is (nearly) species-independent. We also note that the elimination of common distractor strings was particularly important in the performance of our system.

4 Protein Interaction Article Subtask (IAS)

We submitted three runs for the IAS subtask. These were generated by training machine-learning-based classifiers on linguistic and semantic features extracted from the training data. The most distinctive aspects of our approach to this task were 1) *Use of semantic features* and 2) *An attempt to balance the training set*. We noted a large discrepancy between our results on the training data and our results on the test data that we suspect reflects conceptual drift in the document collection. We discuss this at length at the end of this section.

We utilized the WEKA toolkit [33] to constuct the ML-based classifiers. The features employed were n-grams (with n ranging from one to five) of stemmed words and matches to OpenDMAP patterns indicative of protein-protein interaction mentions (See Section 5). Table 12 summarizes the characteristics of the three classifiers that we built.

Balancing positives and negatives in the training data: For our third submission, we balanced the number of positive and negative training abstracts. (There were 3536 positive abstracts, compared with 1959 negative abstracts, in the training set.) We built an additional set of negative abstracts with characteristics similar to the positive abstracts by compiling a collection of verbs that are often used to describe genetic interactions (e.g. *enhance, express,* and *transactivate* and then querying MEDLINE with those terms. We narrowed that set down further by applying our Run-1 classifier to it, thus identifying abstracts which did not discuss protein-protein interaction, and added those articles to create an expanded training set with a 1:1 ratio of positive to negative abstracts. We trained a new classifier on this expanded training set and applied it to the test data to generate this submission.

Results: Our three classifiers achieved F-measures roughly equivalent to one another, and above, but

Table 12: The three classifiers used for the IAS subtask. *IG threshold* is the information gain feature selection threshold.

Name	Classifier		IG threshold
Run 1	SVM	RBF kernel, complexity factor 100, gamma 0.001	.0001
Run 2	Naïve Bayes	kernel estimation enabled	.001
Run 3	SVM with balanced +/-	RBF kernel, complexity factor 100, gamma 0.001	.0001

Table 13: IAS performance compared to the mean and median.

Run	Precision	Recall	F-measure	Accuracy	AUC
Run 1	0.699	0.853	0.768	0.743	0.754
Run 2	0.609	0.941	0.739	0.688	0.562
Run 3	0.706	0.813	0.756	0.737	0.752
Overall mean (standard deviation)	0.664 (0.081)	0.764 (0.193)	0.687 (0.104)	0.671 (0.064)	0.735 (0.074)
Median	0.677	0.851	0.722	0.668	0.752

within one standard deviation of, the overall mean. As in our cross-validation experiments on the training data, our first run achieved the best F-measure, but the difference in F-measures between the three are relatively small. The SVM classifiers (Runs 1 and 3) appear to achieve a higher precision and lower recall than the NB classifier, a characteristic that we have noticed in other document classification work where we compared these classification algorithms [2].

Discussion: We note that our IAS classifiers achieved much higher performance in cross validation on training data than on the test data. For example, our Run-1 classifier achieved a Precision of 0.951, a Recall of 0.945, and an F-measure of 0.948 in 10-fold cross validation of training data; the F-measure achieved in cross-validation was approximately 20% higher than that achieved on the test data. Cross-validation experiments are designed to minimize the effects of over-fitting, and our past experiences suggest that it is typically more successful than was indicated by this experiment. This suggests that a difference exists between the data compiled for the training set and the test set.

We analyzed the corpora and found that the publication years of the articles in the different sets showed that all of the positive training articles were published in either 2005 or 2006, while the negative training articles came from a wider distribution of publication years, centered around 2001. Only about 10% of the negative training articles were published in 2005 or 2006, so it is possible that our classifiers discriminated partially based on the publication years (possibly represented in the feature sets, for example, by a bias in the types of experimental procedures mentioned). Our Run-1 system expressed a bias toward positive classification on the test set (458 positive classifications and 292 negative classifications), where 91% of the articles were published in 2006.

A. Cohen et al. (2004) noted a similar phenomenon in the TREC 2004 Genomics track data. Our short analysis supports the findings of their more extensive study. The difference that they noted was substantially smaller than the one that we report here—about 12%, versus the approximately 20% that we report—suggesting that the training/test data for BioCreative might represent a good data set for working on this problem.

While this apparent publication year bias appears to be an issue with the construction of the training and test corpora, it represents a real-world problem that needs to be dealt with if we are to develop truly useful machine-learning-based document classification systems. Since ideally we would train on currently available data, and apply our systems to literature as it is published, we would require that systems not be affected by this type of bias. A concept-based approach where terms are recognized and mapped to an ontology, as opposed to a purely linguistic-based approach, as we employed for these classifiers, might help avoid over-fitting of classifiers to development data sets. For example, we found that terms describing experimental approaches for detecting protein-protein interactions (e.g., *yeast two hybrid*, *two dimensional gel electrophoresis*, *coimmunoprecipitation*, and *MALDI-TOF*) were among the most important features in discriminating positive from negative articles. The useful information in these features is not the mention of a specific experimental method, but the fact that a technique for recognizing protein-protein interactions was mentioned. (As one reviewer suggested, this points out the value of having a knowledge model that reflects the curation criteria for the reference databases, since they only contain experimentally confirmed interactions.) This is consistent with the hypothesis (advanced by us and others elsewhere, e.g. [2, 3, 6]) that a better approach when training classifiers is to attempt to map words to their underlying concepts. Using this approach, we hypothesize that future systems would be more scalable and robust.

5 Protein Interaction Pairs Subtask (IPS)

For the IPS subtask we used the OpenDMAP (Open source Direct Memory Access Parsing) semantic parser. As is typical for semantic parsers using manually-constructed grammars, our system is geared towards optimizing precision. The procedure begins with preprocessing the HTML, then moves to species recognition, entity tagging, and part of speech tagging, followed by extraction of protein-protein interactions. Our approach to detecting interacting protein pairs relies heavily on the systems generated for the GM and GN tasks.

5.1 Preprocessing

HTML Parsing: The HTML parser developed to process the raw HTML documents was an extension of a similar parser developed for the TREC Genomics 2006 task [3]. Embedded HTML tags are removed, and images representing Greek characters are converted to ASCII strings. The title, abstract, paragraphs, sentences, section headings, and sub-section headings were extracted for each document. Document sections are inferred based on the section heading text. Sentence boundaries are detected using the LingPipe sentence chunker [4]. Sentences are mapped back to the original HTML using a dynamic programming approach.

Gene Mention Tagging: We used a variant of the system developed for the GM task to tag genes. For the IPS task, the outputs of ABNER [28] (both models) and LingPipe [4] (BioCreative04 model) were combined using the *overlapping filter* (See Section 2).

Part of Speech Tagging was done with the GENIA POS Tagger [31].

Species Classification was done with a modified dictionary search. The dictionary was constructed from the intersection of words from the NCBI `names.dmp` file (a list of all known scientific names and synonyms for organisms) and the set of NCBI taxonomy identifiers present in the IPS training set. These words were then combined into a single regular expression pattern for each species. Some filtering of false positive species mentions was achieved by evaluating bigrams present in the flanking regions of each species mention. Each species detected in an article was given a score based on the number of times it appeared and results of its flanking region evaluation, and a ranked list of species for a given article was returned. The BioCreative 2006 PPI training set was used to create a list of bigrams present in the flanking region (±50 character positions) of each species pattern match. These “indicator bigrams” were each assigned a log-odds score corresponding with the formula:

p(bigram occurs in the flanking region of a true positive dictionary match) / p(bigram occurs in the flanking region of a false positive dictionary match)

The species patterns found in each test article were given scores according to the sum of all log-odds scores for indicator bigrams found in the flanking region. The total score for a given species classification for a single article was calculated by summing all individual pattern match scores. Once scored, the species for a given document are returned in rank order. We experimented with the optimal number of species results to return and found the best results when the maximum number of species returned from the ranked list was two.

5.2 Gene Mention Normalization

Gene Lexicon Construction: Dictionaries were constructed for the IPS task for each species that was observed in the training data by extracting information from the `uniprot_light_table_updated.txt` file supplied by the BioCreative organizers.

Gene Mention Normalization: Each gene mention was normalized using the procedure described above for the GN task, using the dictionary for the apparent species. We experimented with the optimal number of normalized identifiers to return and found the best results when we limited the output to one normalized entry per gene mention in text.

5.3 OpenDMAP and Conceptual Patterns

OpenDMAP patterns are written in a context-free syntax. Non-terminal elements are defined in a Protégé ontology. A simple example of an OpenDMAP pattern for the IPS task looks like:

{*interaction*} → *[interactor1] interacts with [interactor2];*

. . . where elements in {braces} represent classes in the ontology, elements in [brackets] correspond to slots of the class on the left-hand side of the pattern, and bare strings are terminals. The slots are constrained in the ontology to have specific features; for the IPS task, the slot elements [interactor1] and [interactor2] are constrained to be proteins. The output is sentences in which OpenDMAP found text matching a protein interaction pattern, as well as the entities involved in the interaction.

OpenDMAP patterns allow for recursion and for the free mixing of terminals and non-terminals. For instance, the following patterns:

{*interaction*} → {*interact-noun*} {*preposition*} {*determiner*}*?* {*protein-list*} *with* {*protein-list*}*;*
{*protein-list*} → *[interactor1] and [interactor2];*

. . . match the bolded text in *The present report examines protein-protein* ***interaction of NMT1 and NMT2 with m-calpain and caspase-3*** *in human colorectal adenocarcinoma tissues and HC-CLs* (PMID 16530191) and returns the four interacting pairs NMT1/m-calpain, NMT1/caspase-3, NMT2/m-calpain, and NMT2/caspase-3.

We used a variety of discovery procedures to build the grammar, including scheduled elicitation sessions with "native speakers" (scientists with expertise in biology) and examination of corpora for frequently-occurring ngrams and frequently-occurring strings between protein mentions [25]. We used the BioCreative 2006 IPS, ISS and IAS training data, the PICorpus[4] [1, 15], material generated by Jörg Hakenberg [23][5] and Anna Veuthey, and the Prodisen corpus[6]. We tuned the system using a 40,000-sentence subset of the IPS training data. The final grammar consisted of 67 rules. It handles verbs, nominalizations, and various forms of conjunction, but not negation.

5.4 Results

There was a marked difference between our performance on the training data and on the test data. Our results on the training data were P = 0.364, R = 0.044, and F = 0.078, returning 385 pairs. However, we achieved recall as high as 0.31 on the test data (seven times higher than on the training data), and recall higher than the median on 2 of 5 measures (see Tables 14 and 15). Our F-measure was above the median more often than it was below it.

6 Protein Interaction Sentences Subtask (ISS)

We modelled the ISS subtask as a summarization task, using an approach similar to the Edmundsonian paradigm: we created a scoring scheme to rate sentences as either containing an interaction mention, or not. This approach has been used for selecting candidate GeneRIFs from Medline abstracts [18].

[4] Available at `http://bionlp.sourceforge.net/`

[5] Available at `http://www2.informatik.hu-berlin.de/~hakenber/`

[6] Available at `http://www.pdg.cnb.uam.es/martink/PRODISEN/`

Table 14: Comparison of interaction pairs results

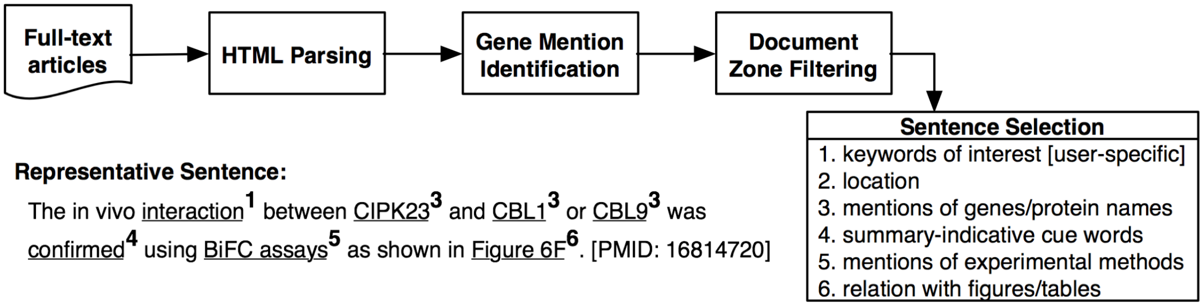
	calculated by interaction			calculated by article		
	P	R	F	P	R	F
Run 1	0.38	0.06	0.11	0.39	0.31	0.29
Median	0.06	0.11	0.07	0.08	0.20	0.08

Table 15: Comparison of normalization

	calculated by interactor			calculated by article			calculated by article with interactions		
	P	R	F	P	R	F	P	R	F
Run 1	0.57	0.12	0.19	0.15	0.13	0.13	0.56	0.46	0.48
Median	0.18	0.25	0.19	0.16	0.28	0.17	0.21	0.39	0.24

Task Data: Table 16 shows the size of the development and blind test data sets. The training set includes a total of 29 full-text articles with 53 gold-standard sentences selected by IntAct [7] and MINT [8] database curators. On average, in the development set, there are approximately two sentences per article, which is markedly smaller than the average number of sentences (5.5) per article in the test set. We did not make use of the number of interaction sentences per article in the development set; systems that did would be likely to undergenerate.

Sentence Selection: Each candidate interaction sentence is scored based on criteria which differ depending on the location of the sentence in the document[9] (Table 17). In order to be scored, the sentence first must meet certain eligibility requirements.


Full-text articles
HTML Parsing
Gene Mention Identification
Document Zone Filtering
Sentence Selection
1. keywords of interest [user-specific]
2. location
3. mentions of genes/protein names
4. summary-indicative cue words
5. mentions of experimental methods
6. relation with figures/tables
Representative Sentence:

The in vivo interaction[1] between CIPK23[3] and CBL1[3] or CBL9[3] was confirmed[4] using BiFC assays[5] as shown in Figure 6F[6]. [PMID: 16814720]

Figure 1: General system design for finding protein interaction sentences from full-text articles. A sample gold standard sentence is shown on the bottom left with key scoring components underlined and numbered according to the corresponding sentence selection category.

Scoreable Features: Our system scores each sentence in a full-text article with respect to these features (Figure 1):

1. Frequent words: The frequent words in the gold standard are all related to protein-protein interaction. For instance, the word *interact* and the phrase *interaction of* are the most frequent unigram and bigram, respectively.
2. Location: We found most gold-standard passages in the *Results* section, and few in the *Title*, *Abstract* or *Introduction* sections. Some sections never yielded a sentence.
3. Mentions of gene/protein names: Since the sentences make assertions about protein-protein interaction, protein mentions are necessary in these sentences.

[7] `http://www.ebi.ac.uk/intact`

[8] `http://mint.bio.uniroma2.it/mint/`

[9] Section-specific usefulness and error rates have been noted in other BioNLP application areas, e.g. [30].

Table 16: BioCreative II protein interaction sentences (ISS) task: development and test data sets.

Data set	articles	sentences	interaction sentences/article
Development (July release)	9	24	2.67
Development (Sept. release)	20	39	1.95
Development (Sum)	29	53	1.83
Blind test	358	1,978	5.53

Table 17: **Scoring requirements** P: Has positive cue words, N: Does not have negative cue words, G: has >0 gene mentions, X: has experimental methods, I: has interaction key word; * If a sentence includes a reference to a figure or table, the score for the caption is added to the score for the sentence.

	Requires					Scored on				
Location	P	N	G	X	I	P	N	G	X	I
Abstract		√	√						√	√
Figure/Table Caption		√	√	√	√				√	√
Section/Subsection Heading		√			√				√	√
Other*		√	√		√	√			√	√

4. Summary-indicative cue words: Words (e.g. *confirm*) or phrases (e.g. *data establish*) that indicate a sentence is likely to be a good interaction sentence.
5. Mentions of experimental methods: Protein-protein interaction detection methods (e.g. *two hybrid array*) are frequently mentioned in the gold-standard passages.
6. Figure/table mention: Many gold-standard passages refer to a table or figure.

Preprocessing: The methods used for HTML parsing and gene name tagging were the same as used for the IPS task (See Section 5). In an attempt to remove false positives prior to processing, we implemented a document zoning filter which excluded sentences associated with certain document sections. The excluded document sections were chosen from manual inspection of some of the training data. The sections include: *Materials and Methods, Acknowledgments, Discussion, Reference, Table of Contents, Disclosures*, and *Glossary.*

Results: We submitted two runs for the ISS task. The runs differed only in the passage length returned for each "interaction sentence." For our Run #1, the returned passage was limited to a single sentence. This restriction was loosened for Run #2, permitting multiple consecutive high-scoring sentences to be returned.

Our results show that loosening the passage length restriction permitted the extraction of 39.2% more passages that had been pre-selected by the human curators when compared to our single-sentence run (Table 18). This suggests that informative sentences regarding protein interactions in full text are likely to be found in close proximity. This contrasts with the case of abstracts, in which such sentences tend to be found at opposite ends of the text [18]. Note that we made no attempt to rank our outputs.

7 Discussion

Preliminary results suggest that the BioCreative 2006 PPI materials might be a fruitful data set for investigating the issues of conceptual drift raised by [8].

A major goal of our work on this shared task was to extend the OpenDMAP semantic parser. We did so, incorporating a number of third-party linguistic and semantic analysis tools without sur-

Table 18: ISS results. Runs 1 and 2 are our submissions. *Passages*, the total number of passages evaluated; *TP*, the number of evaluated passages that were pre-selected by human curators; *Unique*, the number of unique passages evaluated. *U_TP*, the number of unique passages that were pre-selected; *MRR*, mean reciprocal rank of the correct passages.

Run	Passages	TP	Unique	U_TP	TP/Passages	U_TP/Unique	MRR
Run #1	372	51	361	51	13.71	14.13	1.0
Run #2	372	71	361	70	19.09	19.39	1.0

rendering an essential characteristic of the DMAP paradigm: complete integration of semantic and linguistic knowledge, without segregating lexical and domain knowledge into separate components.

We used UIMA [11, 20] as a framework for integrating the various software components used throughout our BioCreative 2006 submissions. For each major component, a UIMA wrapper was created so that it could be plugged into the system. For the GM task, a UIMA wrapper was created for each gene tagger. A component for reading in the document collection was also created, as was a component for outputting the results into the format required by the *alt_eval.pl* script. The output component was also crucial for converting the annotation spans created by the taggers into the somewhat idiosyncratic output format required by the competition organizers. Using the UIMA framework enabled our system to quickly convert between the two different filters, *consensus* and *overlapping*, by simply swapping out the components, and to evaluate their effects quite quickly.

By using a standardized framework, we were not only able to distribute the tasks of development with the assurance that the pieces would work in concert once combined, but we were also able to design our systems in such a way that as they became successively more complicated, evaluation remained quick, easy, and modular. Not only was it possible to incorporate infrastructure constructed expressly for the BioCreative tasks, but it was just as easy able to utilize external tools developed prior to the BioCreative tasks and/or by third-parties. This allowed us to benefit from LingPipe, Schwartz and Hearst's abbreviation-defining algorithm, ABNER, KeX, ABGene, and the GENIA POS tagger (op cit). Utilizing this framework provided not only a robust development architecture and production-ready execution environment, but also a tremendous time savings.

8 Acknowledgments

We thank the two anonymous BioCreative reviewers for their insightful feedback. This work was supported by NIH grant R01-LM008111 to Larry Hunter.

References

[1] Christian Blaschke, Miguel A. Andrade, Christos Ouzounis, and Alfonso Valencia. Automatic extraction of biological information from scientific text: protein-protein interactions. In *Intelligent Systems for Molecular Biology 1999*, pages 60–67, 1999.

[2] Gregory J. Caporaso, William A. Baumgartner, Jr., Bretonnel K. Cohen, Helen L. Johnson, Jesse Paquette, and Lawrence Hunter. Concept recognition and the TREC Genomics tasks. In *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, 2005.

[3] JG Caporaso, WA Baumgartner, Jr., H Kim, Z Lu, HL Johnson, O Medvedeva, A Lindemann, LM Fox, EK White, KB Cohen, and L Hunter. Concept recognition, information retrieval, and machine learning in genomics question answering. In *Proceedings of The Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.

[4] Bob Carpenter. Phrasal queries with LingPipe and Lucene: ad hoc genomics text retrieval. In *Proceedings of the 13th annual Text Retrieval Conference*, 2004.

[5] Lifeng Chen, Hongfang Liu, and Carol Friedman. Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics*, 21(2):248–56, 2005.

[6] A. M. Cohen. Unsupervised gene/protein entity normalization using automatically extracted dictionaries. *Proceedings of the BioLINK2005 Workshop Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, pages 17–24, 2005.

[7] Aaron M. Cohen. Unsupervised gene/protein named entity normalization using automatically extracted dictionaries. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, pages 17–24, 2005.

[8] Aaron M. Cohen, Ravi Teja Bhupatiraju, and William R. Hersh. Feature generation, feature selection, classifiers, and conceptual drift for biomedical document triage. In *Proceedings of the 13th Text Retrieval Conference*, 2004.

[9] KB Cohen, AE Dolbey, GK Acquaah-Mensah, and L Hunter. Contrast and variability in gene names. In *Proceedings of ACL Workshop on Natural Language Processing in the Biomedical Domain*, pages 14–20, 2002.

[10] H Fang, K Murphy, Y Jin, J S Kim, and P S White. Human gene name normalization using text matching with automatically extracted synonym dictionaries. In *Proceedings of the BioLNP Workshop on Linking Natural Language Processing and Biology*, pages 41–48, 2006.

[11] David Ferrucci and Adam Lally. UIMA: an architectural approach to unstructured information processing in the corporate. *Nat. Lang. Eng.*, 10(304):327–348, 2004.

[12] W. Fitzgerald. *Building Embedded Conceptual Parsers.* PhD thesis, Northwestern University, 1994.

[13] WN Francis and H Kucera. *Brown Corpus Manual.* Brown University, 1964.

[14] K Fukuda, A Tamura, T Tsunoda, and T Takagi. Toward information extraction: identifying protein names from biological papers. *Pac Symp Biocomput*, pages 707–18, 1998.

[15] Helen L. Johnson, William A. Baumgartner, Jr., Martin Krallinger, K. Bretonnel Cohen, and Lawrence Hunter. Refactoring corpora. In *Proceedings of the BioNLP workshop on linking natural language processing and biology at HLT-NAACL 06*, pages 116–117, 2006.

[16] Shuhei Kinoshita, K. Bretonnel Cohen, Philip V. Ogren, and Lawrence Hunter. BioCreAtIvE task1A: entity identification with a stochastic tagger. *BMC Bioinformatics*, 6 Suppl 1(1471-2105 (Electronic)):S4, 2005.

[17] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, pages 24–26, 1987.

[18] Z Lu, KB Cohen, and L Hunter. Finding GeneRIFs via Gene Ontology annotations. *Pac Symp Biocomput*, pages 52–63, 2006.

[19] Zhiyong Lu. *Text mining on GeneRIFs.* PhD thesis, University of Colorado School of Medicine, 2007.

[20] R. Mack, S. Mukherjea, A. Soffer, N. Uramoto, E. Brown, A. Coden, J. Cooper, A. Inokuchi, B. Iyer, Y. Mass, H. Matsuzawa, and L.V. Subramaniam. Text analytics for life science using the Unstructured Information Management Architecture. *IBM Systems Journal*, 43:490–515, 2004.

[21] C. E. Martin. *Direct Memory Access Parsing*. PhD thesis, Yale University, 1991.

[22] AA Morgan, B Wellner, JB Colombe, R Arens, ME Colosimo, and L Hirschman. Evaluating the automatic mapping of human gene and protein mentions to unique identifiers. *Pac Symp Biocomput*, pages 281–291, 2007.

[23] Conrad Plake, Joerg Hakenberg, and Ulf Leser. Optimizing syntax patterns for discovering protein-protein interactions. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 195–201, New York, NY, USA, 2005. ACM Press.

[24] MF Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[25] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proc. 40th annual meeting of the Association for Computational Linguistics*, pages 41–47, 2002.

[26] CK Riesbeck. From conceptual analyzer to Direct Memory Access Parsing: an overview. In NE Sharkey, editor, *Advances in Cognitive Sciences*. Ellis Horwood Limited, 1986.

[27] Ariel S Schwartz and Marti A Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. *Pac Symp Biocomput*, pages 451–62, 2003.

[28] Burr Settles. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–2, 2005.

[29] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–32, 2002.

[30] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in full text articles. In *Natural language processing in the biomedical domain*, pages 9–13. Association for Computational Linguistics, 2002.

[31] Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsuji. Developing a robust part-of-speech tagger for biomedical text. In *Advances in informatics—10th Panhellenic Conference on Informatics*, pages 382–392, 2005.

[32] O Tuason, L Chen, H Liu, J A Blake, and C Friedman. Biological nomenclatures: a source of lexical knowledge and ambiguity. *Pac Symp Biocomput*, pages 238–49, 2004.

[33] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, June 2005.

[34] Alexander Yeh, Alexander Morgan, Marc Colosimo, and Lynette Hirschman. BioCreAtIvE task 1A: gene mention finding evaluation. *BMC Bioinformatics*, 6 Suppl 1(1471-2105 (Electronic)):S2, 2005.